Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Comparison of VGG16 and VGG19 CNN Architecture for Rice Disease Image Classification

I Gede Iwan Sudipa[1*], Nuratika[2], I Putu Agus Eka Darma Udayana[3], I Made Subrata Sandhiyasa[4], Putu Mahesa Kama Artha[5]

[1(*)]*Department of Informatics, Faculty of Technology and Informatics, Institute of Business and Technology Indonesia, Denpasar, Indonesia*
[2]*Department of Informatics, Faculty of Technology and Informatics, Institute of Business and Technology Indonesia, Denpasar, Indonesia*
[3]*Master of Informatics, Graduate School, Institute of Business and Technology Indonesia, Denpasar, Indonesia*
[4]*Department of Informatics, Faculty of Technology and Informatics, Institute of Business and Technology Indonesia, Denpasar, Indonesia*
[5]*Department of Informatics, Faculty of Technology and Informatics, Institute of Business and Technology Indonesia, Denpasar, Indonesia*
*Corresponding **author**: iwansudipa@instiki.ac.id

ARTICLE INFO

Received: 01 Dec 2025

Revised: 07 Jan 2026

Accepted: 15 Jan 2026

ABSTRACT

The study aims to compare the performance of VGG16 and VGG19 architectures in classifying rice plant disease images using the Convolutional Neural Network (CNN) method. The models were trained using Adam and SGD optimizers with various learning rates, and evaluated using accuracy, precision, recall, f1-score, and confusion matrix metrics. The results show that VGG19 with Adam's optimizer and learning rate 0.001 provides the best performance with a testing accuracy of 97.90% and the lowest validation loss value of 0.2910. VGG16 also showed good results with the highest validation accuracy of 89%, but the results were below VGG19. Based on the confusion matrix, both models accurately recognize some classes, although there are still errors in certain classes. Thus, the results obtained show that VGG19 is superior in terms of training accuracy and stability. The use of data augmentation techniques is expected to reduce the potential for overfitting.

Keywords: Image Classification, Rice Plant Disease, Deep Learning

INTRODUCTION

Rice is a type of cereal plant that has an important role in supporting the economy in Indonesia's agricultural sector. Rice produced by rice is used as one of the staple foods that are widely consumed by the Indonesian people. Therefore, the productivity of rice plants needs to be considered because production failure can cause economic and political disruption[1] . The survey results stated that Indonesia's 2023 rice production reached 53.98 million tons of MDG, down 1.40% compared to 2022, equivalent to 31.10 million tons of rice, down 1.39%. The highest production was recorded in March (8.92 million tons of MDG) and the lowest in December (1.97 million tons of MDG). In January-April 2024, it is estimated that total paddy and rice production reached 18.59 million tons of MDG (down 17.54%) and 10.71 million tons of rice (down 17.52%) respectively compared to the previous year. The increase only occurred in May-August 2023, up 7.03% [2]–[5].

The decline in rice production was caused by the attack of Plant Disturbing Organisms. Plant Disturbing Organisms are all types of organisms that have the potential to cause damage and disturbance to certain rice or secondary crops, which include pests, diseases, and weeds[6] . The attack of Plant Disturbing Organisms has a very bad impact on the condition of rice[7], [8] , which can result in a decrease in production and other material losses.

Common diseases affecting rice plants are caused by bacteria and fungi, such as leaf blight caused by Xanthomonas campestris which is characterized by dry and gray leaves, and fungal diseases such as leaf blast (Magnaporthe grisea), fusarium leaf (Fusarium sp.), midrib blight (Rhizoctonia sp.), and narrow brown leaf spot (Cercospora oryzae) which causes spots or blight on the leaves[9] . Apart from attacking the leaves, diseases can also attack the grain, such as False Smut caused by Ustilaginoidea virens, causing swelling and a decrease in grain quality which has an impact on

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

yield. Some diseases have similar symptoms such as spotting or leaf discoloration, making identification difficult, such as blast, brown spot, and narrow brown spot[10] . This difficulty can cause delays in treatment and exacerbate infection, so the utilization of image-based automation technology[11], [12] in rice disease classification is a potential solution to accelerate and improve the accuracy of diagnosis.

VGG-19 developed by Visual Geometry Group, is a CNN model with 19 layers, consisting of 16 convolutional and 3 fully connected layers, using ReLU activation. With an input of 224×224 pixels and max pooling at the end of each convolution, this model captures complex features through small convolution layers (3×3)[13] . In[14] research to classify fundus by comparing AlexNet, VGG-16, VGG-19, ResNet50 architectures. Research conducted by[15] compares V3, VGG 16, VGG 19, CNN, and ResNet architectures for rice disease detection. Furthermore by[16], [17] which shows the VGG-19 architecture gets the best accuracy results with a value of 90%. Another study was conducted by[18] to classify banana ripeness by comparing VGG-19 and ResNet architectures. The best accuracy results were obtained by VGG-19 with a value of 100%. Research related to leaf disease detection conducted by[19], [20] with VGG-19 architecture and ADAM optimizer obtained an accuracy of 96.5%. Meanwhile, research conducted by[21] tested the performance of the VGG19 architecture for image classification. The results showed that VGG-19 achieved accuracy results of up to 99.72%.

While the majority of previous studies have consistently ranked VGG-19 as the superior architecture, there are empirical inconsistencies in the findings that prove the superiority of VGG-16[22]–[25] . This indicates that increasing the network depth of VGG-19 does not automatically guarantee better accuracy, leading to uncertainty about the effectiveness of the model. With this in mind, this study contributes through a head-to-head comparative analysis between VGG-16 and VGG-19 to validate which architecture is most optimally applied to rice disease datasets, while clarifying the debate on the effect of layer depth on classification performance.

Based on the results of the previous research above, it shows that the VGG-19 architecture has the best performance in carrying out classification tasks on image data. However, the latest research shows that the performance of VGG-16 is superior to VGG-19. Thus, this research has a contribution in applying both architectures to find out which architecture is superior and has higher accuracy.

METHODS

The first stage in this research is data collection, secondary data techniques are used where the data used is taken from public data on kaggle.com as much as 1,426 rice plant image data. The preprocessing process is carried out to prepare the data to be ready for use, including cropping to cut out parts of the image that are not needed, and resizing is useful for adjusting the image size to be uniform. After that, the data is divided through a split dataset by separating the data into training data, and test data. CNN model training is performed using the processed dataset, and hyperparameter optimization is required to achieve optimal results. Finally, the performance of the model is evaluated using confusion matrix.

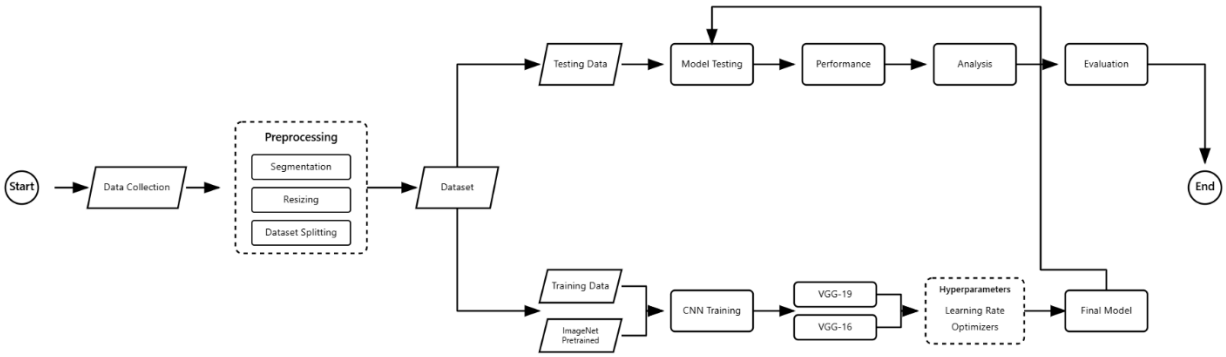

Figure: 1 Research Stage

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Data Collection

Data collection in this study used secondary data collection techniques. The author uses public data that has been processed and collected from the Kaggle site related to rice-disease data1. From the site, image data of rice plants that have been available in the form of datasets and used as the main material to support the process of analysis and model testing, a total of 1,426 images consisting of 9 classes, namely Bacterial Leaf Blight (BLB) 138 images, Brown Planthopper (BPH) 71 images, Brown Spot 111 images, False Smute 93, Healthy plant 234 images, Hispa 73 images, Neck Blast 286, Sheath Blight Rot 219, and Stemborer 201 images.

Preprocessing

The preprocessing stage in a Convolutional Neural Network (CNN) aims to simplify the image to make it easier for the model to process. This process involves techniques such as segmentation and resize to ensure the image has a uniform format and is suitable for the needs of the model. By preprocessing the data inputted into the model can be optimally processed, thereby improving the accuracy and performance of the model.

Segmentation

The segmentation stage is carried out with contour detection to perform automatic cropping which aims to remove parts of the original image that do not have important information and only leave the necessary parts. The segmentation stage uses thresholding to distinguish the affected area from the unnecessary background.

Resizing

In the resizing stage, images that have different sizes will be changed so that they become the same size and consistent, so that it can facilitate the process of feature extraction and image recognition. The image size is changed to 224 x 224 pixels to match the input size required by the VGG-19 algorithm. To run the resizing, researchers used python code.

Split Dataset

The split dataset process is done by dividing the data into training data and testing data. Training data is used to train and introduce the model before testing the testing data. While testing data is used to evaluate the final performance of the model with new data that has never been used before. This research uses data as much as 1,426 image data which is divided into 9 classes of disease in rice plants. From this data, the author divides the data in a ratio of 70%: 15%: 15%. Where 70% is the data used for the training process, 15% is the data used for the testing process and 15% for validation data.

Table 1: Split Dataset

Class	Total Image	Training Data	Testing Data	Validation Data
Bacterial Leaf Blight (BLB)	138	96	20	20
Brown Planthopper (BPH)	71	49	11	11
Brown Spot	111	77	17	17
False Smute	93	65	14	14
Healthy plant	234	164	35	35
Hispa	73	51	11	11
Neck Blast	286	200	43	43
Sheath Blight Rot	219	153	33	33
Stemborer	201	141	30	30
Total Dataset	1.426	998	214	214

Model Training

The training process begins with processing the input data, then continues with adjusting the weights in each layer. The model is trained using epochs to improve accuracy and reduce loss values in the training data. Throughout the training, validation data is used to prevent overfitting and ensure the model can perform well on new data. The final

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

model will be evaluated using test data to initiate classification performance. In the training process, hyperparameter optimization is used for the model with the best results. The hyperparameters used are epoch, learning rate, and optimizer. Epoch serves to determine the number of iterations when the entire dataset is processed by the model during training. Learning rate is done to determine how much weight change is applied during the training process. Optimizer is used to optimize the parameters in the model.

Implementation of CNN Method with VGG-19 Architecture

Convolutional Neural Network (CNN) method with VGG-19 architecture consisting of 16 convolutional layers with ReLU activation and 3 fully connected layers, is implemented in this research. The input image of rice plant disease is 224 x 224 pixels in size, with a 3 x 3 kernel for feature extraction, and max pooling using a 2 x 2 filter in each layer to reduce dimensions and improve model efficiency.

Implementation of CNN Method with VGG-16 Architecture

This research uses VGG1-16 architecture consisting of 16 layers, with 13 convolution layers and 3 fully connected layers. The input image has a size of 224x224 pixels, then the number of filters used is 64, 128, 256, and 512, with a 3x3 carnel-sized convolution block with stride 1, followed by a 2x2 max pooling layer with stride 2 which ends with fully connected and softmax.

Model Performance Evaluation

Evaluation is done with Confusion Matrix to assess the performance in the classification task. This method measures classification accuracy, precision, recall and f1-score to provide an overall picture of the model's ability to correctly classify the data. The evaluation process utilizes the original dataset of rice plant diseases, which represents the actual conditions. Adam's algorithm and SGD were also used in this study to perform optimization. The results of this evaluation help understand the strengths and weaknesses of the model, as well as a basis for making improvements if needed.

RESULTS AND DISCUSSION

Implementation of VGG16 Model Training

In this training process, four experimental scenarios were conducted combining two types of optimization algorithms, namely Adam and SGD (Stochastic Gradient Descent), with two variations of the Learning Rate value, namely 0.01 and 0.001. Each combination of optimizer and learning rate was used to evaluate the performance of the model under different conditions to obtain the most optimal training results.

(A) Training using Adam optimizer (Lr 0.001)

During the model training with 10 epochs, the initial accuracy at the first epoch was only 31%, but increased sharply to 90% at the second epoch as the model began to learn effectively through weight adjustment. The training accuracy continued to increase until it reached 100% at the 5th epoch and was stable until the end, while the validation accuracy remained consistent in the range of 86-87%, showing good generalization ability without any indication of overfitting. The training loss value decreased close to zero, and the best val_loss of 0.3423 was achieved at the 9th epoch.

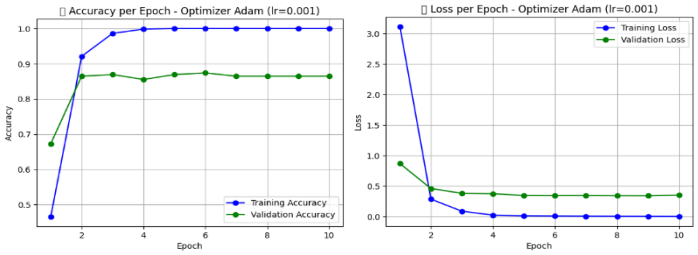

Figure: 2 Diagram of Acc and Loss opt Adam Lr 0.001

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Based on the confusion matrix, most of the predictions are on the main diagonal, indicating accurate classification of most classes, such as class 6 (90 data), class 7 (78), and class 4 (57), although there are some errors such as class 0 6 times. The final evaluation showed an accuracy of 95.56% with a precision of 95.71%, recall of 95.56%, and f1-score of 95.55%. The macro and weighted average values for precision, recall, and f1-score were above 0.95, indicating that the model was able to perform excellent and balanced multi-class classification, even on unbalanced data.

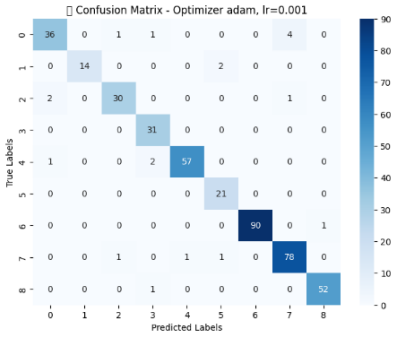

Figure: 3 Confusion Matrix

(B) Training using Adam's optimizer (Lr 0.01)

Based on the training results, the model shows rapid performance improvement from the first to the third epoch, characterized by a decrease in loss and an increase in accuracy in training and validation data. From the 4th to the 10th epoch, the training accuracy reaches 100% and is stable, while the validation accuracy continues to increase until it reaches 88.79%, with the best val_loss of 0.3704 at the 7th epoch. The accuracy and loss graphs show optimal performance, where the training accuracy reaches 99% at the 5th epoch and stabilizes until the end, while the validation accuracy also increases although it fluctuates, then stabilizes at 90%. The significant decrease in loss at the beginning of training shows that the model is able to learn the data pattern effectively without excessive overfitting.

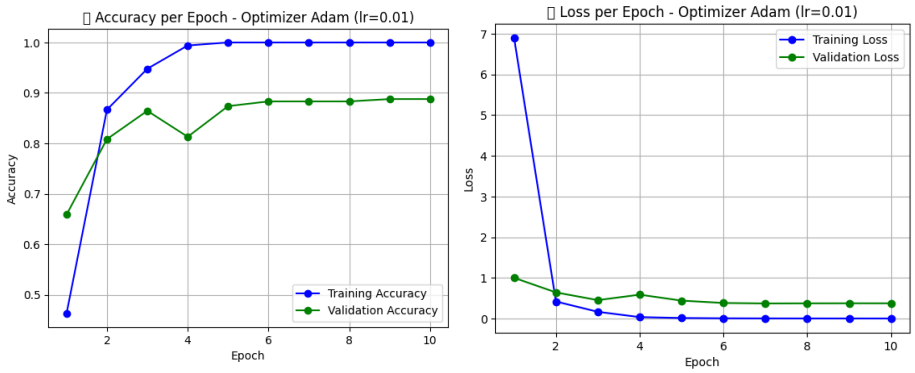

Figure: 4 Diagram of Acc and Loss opt Adam Lr 0.01

The confusion matrix shows that the classification model with Adam optimizer and learning rate 0.01 performs very well against the 9 target classes, characterized by the dominance of high numbers on the main diagonal such as 90 (class 6), 79 (class 7), and 56 (class 4), as well as very small misclassification errors. The model achieved 96.26% accuracy on the test data, with an average precision, recall, and f1-score (macro and weighted) of 0.96, indicating the model's ability to classify each class equally and accurately.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

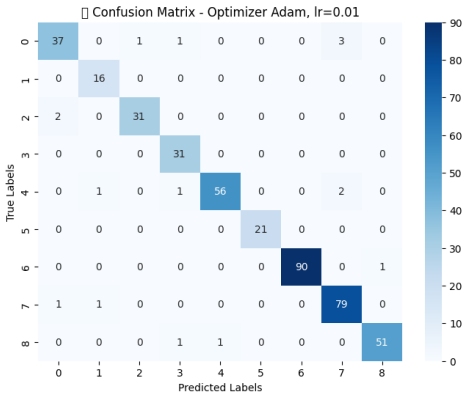

Figure: 5 Confusion Matrix

(C) Training using SGD optimizer (Lr 0.001)

The training results above show stable model performance with high training and validation accuracy throughout the 10 epochs. The training accuracy continued to increase until it reached 94.1% until the 9th epoch and experienced a slight decrease at the 10th epoch to 92.9%. Validation accuracy increased with the highest value of 81.31%. The loss value in the validation data gradually decreased until it reached the lowest value of 0.6141 at the 10th epoch. This shows that the model learned well without any indication of overfitting, and the performance improvement continued during the training process, characterized by the consistent improvement of val_loss value.

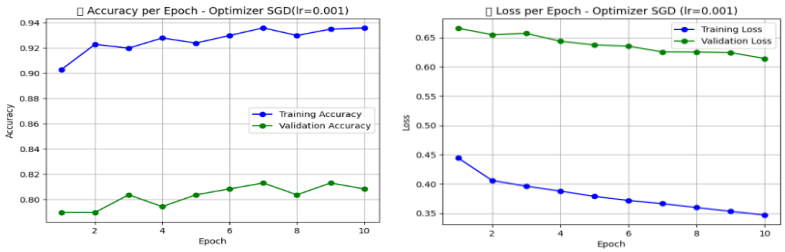

Figure: 6 Acc and Loss diagram (D) Training using SGD optimizer (SGD 0.001)

The confusion matrix results show that the model is quite good at recognizing most numbers (0-8), especially the numbers 6, 4, and 8 which are correctly classified 52, 28, and 22 times respectively. However, the model still has difficulty distinguishing between the numbers 7 and 6, as seen from the misprediction of the number 7 as 6 20 times. Based on the classification report, the model's accuracy on the test data reached 80.84%, with the best performance on the number 6 and fairly accurate classifications on the numbers 1, 4, 5, and 8. In contrast, the model had trouble distinguishing the numbers 0 and 7, reflected in the lower recall value for these classes.

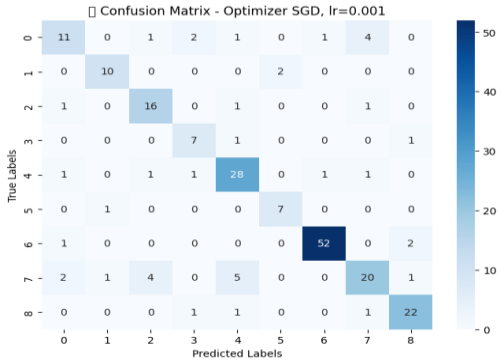

Figure: 7 Confusion Matrix

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

(D) Training using SGD optimizer (Lr 0.01)

The training results show that the model has improved quite well in performance. The training accuracy increased gradually from 86.6% at the 1st epoch to reach 98.0% at the end of the training. The validation accuracy also increased, with the highest value of 85.98% at the 10th epoch. In addition, the val_loss value shows a considerable decrease, from 0.6572 initially to 0.4096 at the end of training. Although there were some fluctuations in the val_loss value and validation accuracy in some epochs, in general the model showed good learning ability without any indication of serious overfitting. This can be seen from the trend of increasing accuracy and decreasing loss in both the training and validation data.

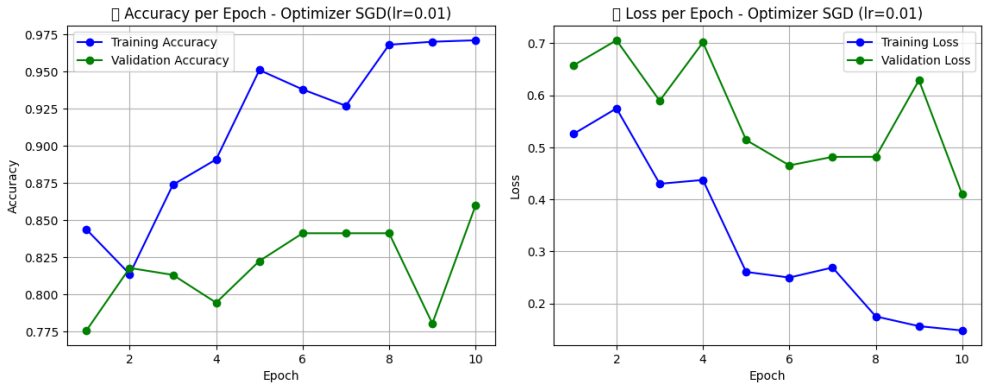

Figure: 8 Acc and Loss Diagram of SGD 0.001

Based on the confusion matrix, the model performed quite well in recognizing numbers, especially 6, 4, and 8, with 6 being correctly classified 54 times. However, there are still mispredictions, such as the number 7 which is often classified as 0 or 8, as well as the number 0 which is sometimes predicted as 4, 6, or 7. The model achieved an accuracy of 85.98% on the test data, with the best performance on the numbers 1 and 6 (f1-score 0.91 and 0.97), and perfect recall on the number 5 although the precision is low (0.73), indicating false predictions to this class. The model's main difficulty is still seen in distinguishing 0 and 7, with lower f1-score in both classes.

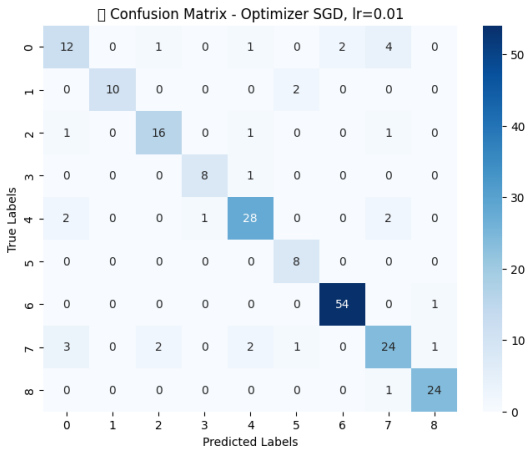

Figure: 9 Confusion Matrix

Model Performance Evaluation

Table: 1 Results of Training Acc and Validation Acc

Model	Optimizer	Learning Rate	Training Accuracy	Validation Accuracy
	Adam	0,001	100%	88%
Dataset	Adam	0,01	100%	89%

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

	SGD	0,001	94%	81%
	SGD	0,01	98%	86%

Table: 2 Scenario Results Confusion Matrix

Model	Optimizer	Learning Rate	Recall	Precission	F1-score	Accuracy
Dataset	Adam	0,001	96%	96%	96%	96%
	Adam	0,01	96%	96%	96%	96%
	SGD	0,001	81%	81%	81%	81%
	SGD	0,01	86%	86%	86%	86%

VGG19 Implementation

(A) Training using Adam's optimizer (Lr 0.001)

Based on the results of 10 epoch training using Adam's optimizer (Lr 0.001), the model performance is quite stable. The training accuracy increased rapidly until it reached 1.00 in the 4th epoch and remained stable, while the validation accuracy also increased at the beginning and then stayed in the range of 0.88 to 0.89. The difference between training and validation accuracy indicates the potential for mild overfitting. The training loss continues to decrease until it approaches zero, while the validation loss, although decreasing slowly, stops and remains in the range of 0.40, indicating that the model is not yet fully capable of generalizing to new data.

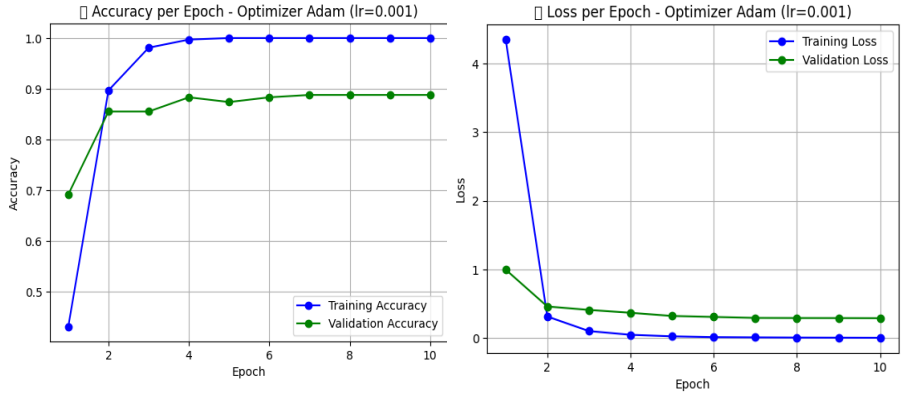

Figure: 10 Diagram of Acc and Loss Adam 0.001

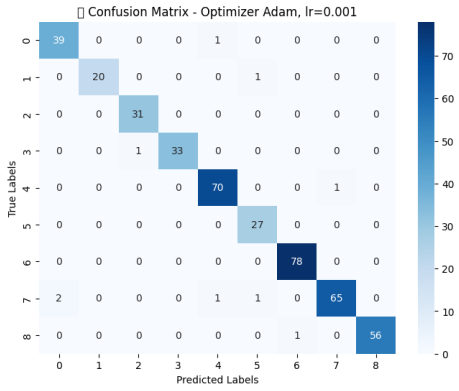

Figure: 11 Confusion matrix

Figure 11 shows the model's excellent performance in classifying numbers, with the numbers 6, 4, 7, and 8 being correctly predicted 78, 70, 65, and 56 times respectively, reflected by the dominance of numbers on the main diagonal

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

of the confusion matrix. Although there were a few errors, such as the number 7 being misclassified as 0 twice, overall the model performed optimally with an accuracy of 97.90% on the test set. Almost all classes have high precision, recall, and f1-score values, and some even achieve perfect values, such as precision 1.00 in numbers 1, 3, and 8, and recall 1.00 in numbers 2, 5, and 6. This indicates the model's excellent ability to recognize various digit classes even though there is little overlap between classes.

(B) Training using Adam's optimizer (Lr 0.01)

From the training results, the model trained using 10 epochs with the Adam optimizer and a learning rate of 0.01 showed that the training accuracy improved rapidly to almost perfect. The validation accuracy also improved and reached around 0.89 at the 10th epoch. However, there is a significant gap between training and validation accuracy, and the validation loss value remains relatively high at around 0.40. This pattern indicates that the model is overfitting, i.e., it is too adaptive to the training data and thus less optimal in generalizing new data.

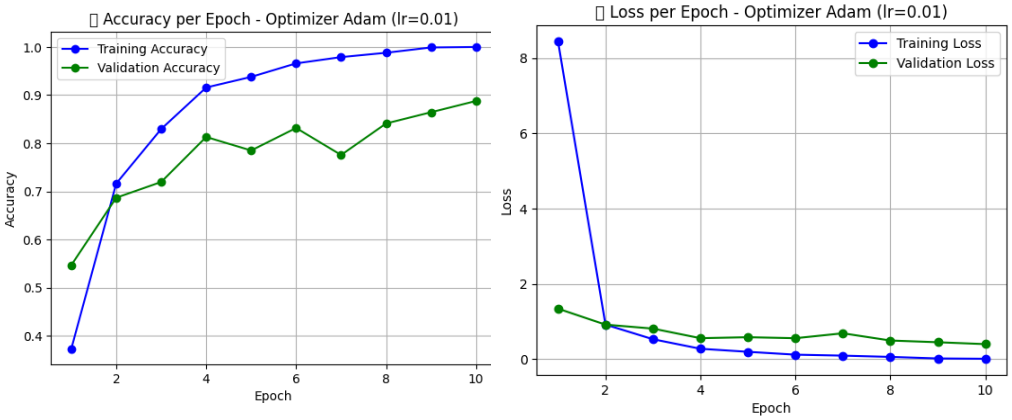

Figure: 12 Diagram of Acc and Loss Adam 0.01

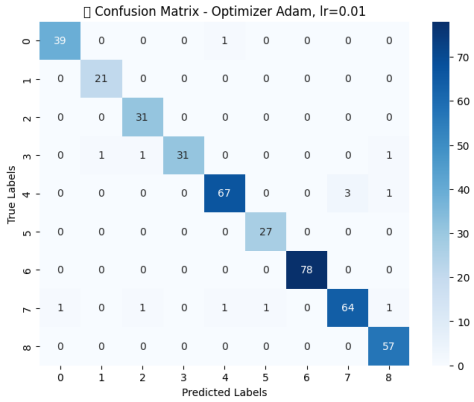

Figure: 13 Confusion Matrix

The confusion matrix in figure 13 shows that the model is able to classify most digits 0 to 8 well after being trained using the Adam optimizer and a learning rate of 0.01, as seen from the correct predictions on the main diagonal such as 6 (78 times), 4 (67 times), and 8 (57 times). Although there were minor errors, such as the number 7 being incorrectly predicted as 0 and 5, and the number 0 sometimes being recognized as 2 or 4, the overall performance of the model was still very good. The classification report recorded an accuracy of 96.96% on the test data, with high precision, recall and f1-score in almost all classes. Numbers 1, 5, and 8 had perfect recall (1.00), while numbers 3 and 6 had perfect precision (1.00), indicating highly accurate predictions. Although there is a slight drop in performance in certain classes such as number 7, overall the model remains consistent and reliable in classification.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

(C) Training using SGD optimizer (Lr 0.001)

From the training results above, the model trained using 10 epochs with SGD optimizer and learning rate 0.001 shows a very different performance between the training data and validation data. The training accuracy reaches 1.00 from the first epoch to the 10th epoch, indicating that the model has fully memorized the training data. However, the validation accuracy only ranges from 0.89 to 0.90, with slight fluctuations. This huge difference indicates severe overfitting. This is reinforced by the loss graph, where the training loss is almost zero throughout the training, while the validation loss remains in the range of 0.38-0.39. This suggests that the model is unable to generalize well to new data and is more focused on getting perfect results on the data it has already seen.

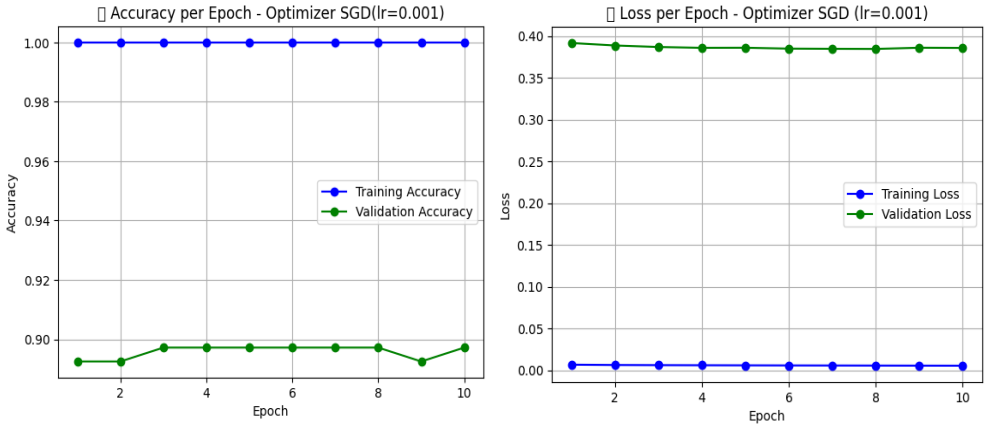

Figure: 14 Acc and Loss Diagram of SGD 0.001

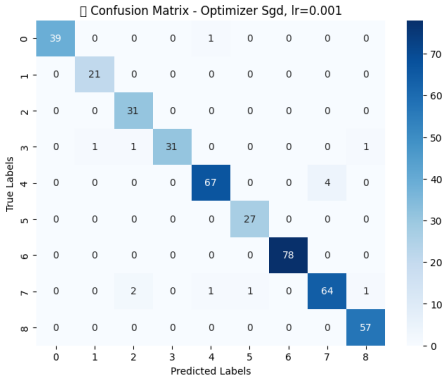

Figure: 15 Confusion Matrix

Based on the confusion matrix in figure 15, the model shows excellent performance in classifying numbers, especially the numbers 6, 4, and 8 which are correctly predicted 78, 67, and 57 times respectively, as seen from the dominance on the main diagonal. Some minor errors still occur, such as the number 7 which is sometimes classified as 2 or 5, and the number 0 which is incorrectly predicted as 4. The classification report results show a high accuracy of 96.96% on the test set, with near-perfect precision, recall, and f1-score in most classes. Perfect precision (1.00) was achieved at 0, 3, and 6, while perfect recall (1.00) was achieved at 1, 2, 5, 6, and 8. Although there was a slight drop in performance at 7 (f1-score 0.93), the model overall still showed highly accurate and reliable classification capabilities.

(D) Training using SGD optimizer (Lr 0.01)

The model trained with the SGD optimizer and learning rate of 0.01 above experienced performance fluctuations during the training process. The training accuracy was consistently high in the range of 0.970 to 0.972 with the highest value achieved at the 2nd epoch. The validation accuracy also showed variations, reaching the highest point of about 0.965 at the 3rd and 6th epochs. The difference between training and validation accuracies is relatively small,

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

indicating that the model does not experience significant overfitting. The training loss tends to decrease although there is a slight spike. Meanwhile, the validation loss shows a sharper fluctuation, especially at the 5th epoch before decreasing again. This pattern shows that the model is good enough, but still has room for improvement to make the results more stable and optimal.

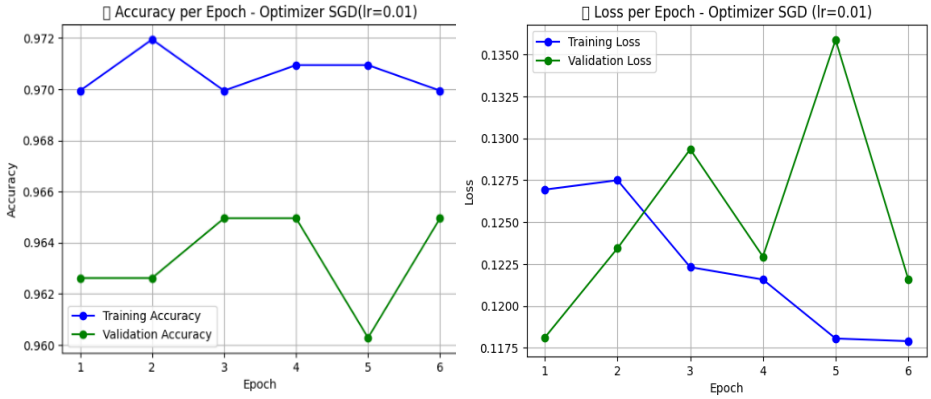

Figure: 16 Acc and Loss Diagram of SGD 0.01

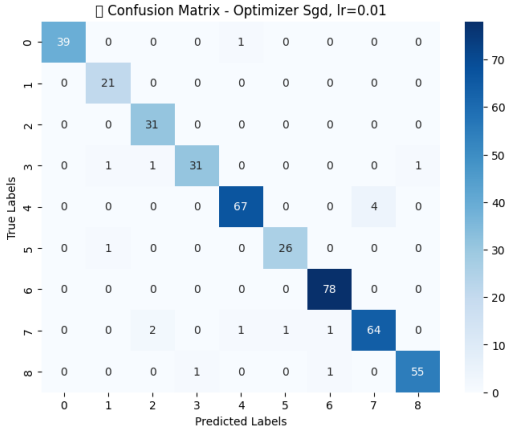

Figure: 17 Confusion Matrix

The confusion matrix shows that the model has a high performance in recognizing most of the digits, especially the numbers 6, 4, and 8 which were correctly predicted 78, 67, and 55 times, respectively. Although there are still errors, such as the number 7 which is sometimes classified as 2, 5, or 6, and the number 0 which is misrecognized as 4, in general the model is able to classify most classes well. Based on the classification report, the model achieved 96.26% accuracy on the test set, with almost perfect precision, recall, and f1-score values in most classes. Number 0 had a precision of 1.00 and an f1-score of 0.99, while numbers 1, 2, and 6 achieved perfect recall (1.00). Although the f1-score of number 7 was slightly lower (0.93), the overall performance of the model was still very good and consistent.

Model performance evaluation

Table: 3 Results of Training Acc and Validation Acc

Model	Optimizer	Learning Rate	Training Accuracy	Validation Accuracy
Dataset	Adam	0,001	100%	89%
	Adam	0,01	100%	89%
	SGD	0,001	100%	90%
	SGD	0,01	97%	96%

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table: 4 Scenario Results Confusion Matrix

Model	Optimizer	Learning Rate	Recall	Precission	F1-score	Accuracy
Dataset	Adam	0,001	98%	98%	98%	98%
	Adam	0,01	97%	97%	97%	97%
	SGD	0,001	97%	97%	97%	97%
	SGD	0,01	96%	96%	96%	96%

In the advanced testing stage, researchers tested the previously trained model using 9 test images representing various types of diseases in rice plants. This test aims to measure the generalization ability of the model to new data that was not used during the training process. The VGG16 model trained using the Adam optimizer with a learning rate of 0.01 successfully predicted two images correctly, namely the Bacterial Leaf Blight (BLB) and Sheath Blight Rot disease images, each with a prediction accuracy rate of 100%. Meanwhile, the VGG19 model trained using Adam's optimizer with a learning rate of 0.001 also successfully predicted two images correctly, namely False Smut and Sheath Blight Rot, both also with 100% prediction accuracy. Based on these results, it can be concluded that both models have the ability to recognize some disease categories well, especially the Sheath Blight Rot disease which is consistently predicted correctly by both models. However, the ability to predict other disease categories is still limited, so it is necessary to improve the performance of the model through development methods such as increasing the amount of training data, using image augmentation techniques, or improving the model architecture parameters.

CONCLUSION

This study concludes that the VGG-19 architecture shows superior performance compared to VGG-16 in rice plant disease classification. Based on the experimental results, the best configuration was achieved by VGG-19 using the Adam optimizer with a learning rate of 0.001, which resulted in the highest test accuracy of 97.90% and an average F1-score of 98%. This outperformed the optimal performance of VGG-16 (Adam, LR 0.01) which only achieved 96% accuracy. In addition, VGG-19 showed better generalization stability with the lowest validation loss value (0.2910) compared to VGG-16 (0.3704), minimizing the risk of overfitting. Analysis of the optimizer confirmed that Adam was consistently more effective than SGD, with SGD yielding significantly lower accuracy (81-86%) on both architectures. Although VGG-16 offers less computational efficiency, VGG-19 proved to be more reliable in capturing complex features in rice disease images, VGG-19 is recommended as the most appropriate architecture model to be applied to rice disease detection systems.

REFERENCES

[1] M. Rondhi, A. Fatikhul Khasan, Y. Mori, and T. Kondo, "Assessing the role of the perceived impact of climate change on national adaptation policy: The case of rice farming in Indonesia," *Land*, vol. 8, no. 5, p. 81, 2019, doi: https://doi.org/10.3390/land8050081.

[2] M. Salam *et al.*, "Determinant factors affecting farmers' income of rice farming in Indonesia," in *IOP Conference Series: Earth and Environmental Science*, 2019, vol. 343, no. 1, p. 12115. doi: 10.1088/1755-1315/343/1/012115.

[3] I. G. A. R. Gotama, I. G. I. Sudipa, A. A. G. R. W. Brahma, M. S. Ariantini, and D. A. P. Wulandari, "Enhanced Stacked GRU Model for Monthly Rice Production Forecasting in Bali Province," *Sink. J. dan Penelit. Tek. Inform.*, vol. 10, no. 1, pp. 652–664, 2026, doi: https://doi.org/10.33395/sinkron.v10i1.15715.

[4] I. K. A. Wiguna, I. G. I. Sudipa, N. P. S. Meinarni, K. J. Atmaja, and A. A. G. Ekayana, "Fuzzy Time Series Chen Model for Dual-Commodity Agricultural Forecasting: Evidence from Indonesia's Rice and Corn Production," *Sink. J. dan Penelit. Tek. Inform.*, vol. 10, no. 1, pp. 468–480, 2026, doi: https://doi.org/10.33395/sinkron.v10i1.15584.

[5] I. G. I. Sudipa, K. M. Aman, I. M. Subrata, K. J. A. Sandhiyasa, and I. G. Sudiantara, "Predictive Time-Series Modelling of Rice Price Fluctuations in East Nusa Tenggara Using ARIMAX: A Data Driven Case Study Regional Agricultural Market Dynamics," *Power Syst. Technol.*, vol. 48, no. 4, pp. 111–123, 2024, doi: 10.52783/pst.972.

[6] M. Y. Mohammad, A. S. S. Jahan, V. Sujarajini, and H. M. Haniffa, "Comprehensive review on types of pest attacks in paddy cultivation and botanical control measures," *Crop. Forage Turfgrass Manag.*, vol. 11, no. 1, p. e70026, 2025, doi: https://doi.org/10.1002/cft2.70026.

[7] S. Fahad *et al.*, "Bio-based integrated pest management in rice: An agro-ecosystems friendly approach for agricultural sustainability," *J. Saudi Soc. Agric. Sci.*, vol. 20, no. 2, pp. 94–102, 2021, doi:

https://doi.org/10.1016/j.jssas.2020.12.004.

[8] I. Vagelas, G. Michail, and P. Madesis, *Pests and diseases: A global threat to plants*. Apple Academic Press, 2024.

[9] N. Jiang *et al.*, "Resistance genes and their interactions with bacterial blight/leaf streak pathogens (Xanthomonas oryzae) in rice (Oryza sativa L.)—an updated review," *Rice*, vol. 13, no. 1, p. 3, 2020, doi: https://doi.org/10.1186/s12284-019-0358-y.

[10] J. Shrestha, M. Kandel, S. Subedi, and K. K. Shah, "Role of nutrients in rice (Oryza sativa L.): A review," *Agrica*, vol. 9, no. 1, pp. 53–62, 2020, doi: https://doi.org/10.5958/2394-448X.2020.00008.5.

[11] I. P. A. T. Wahyudi, I. G. I. Sudipa, L. G. B. Libraeni, M. L. Radhitya, and I. M. D. P. Asana, "Performance Comparison of MobileNet and EfficientNet Architectures in Automatic Classification of Bacterial Colonies," *Indones. J. Data Sci.*, vol. 6, no. 2, pp. 333–342, 2025, doi: https://doi.org/10.56705/ijodas.v6i2.218.

[12] G. R. Baihaqi, S. R. Shalsadilla, and M. K. Argaputri, "Accuracy Improvement of Convolutional Neural Network with Ghost Weight Normalization for Pneumonia Classification," *J. Galaksi*, vol. 1, no. 3, pp. 143–152, 2024, doi: https://doi.org/10.70103/galaksi.v1i3.35.

[13] R. Sinta, Jasril, M. Irsyad, F. Yanto, and S. Jaya, "Klasifikasi Citra Penyakit Daun Tanaman Padi Menggunakan CNN dengan Arsitektur VGG-19," *J. Ekon. Vol. 18, Nomor 1 Maret201*, vol. 2, no. 1, pp. 41–49, 2020.

[14] A. T. Khalaf and S. K. Abdulateef, "Multi-Class of Retinal Diseases Classification via Deep Learning Techniques Based on Fundus Images," *Samarra J. Pure Appl. Sci.*, vol. 6, no. 3, pp. 274–286, 2024, doi: https://doi.org/10.54153/sjpas.2024.v6i3.881.

[15] S. R. Shah, S. Qadri, H. Bibi, S. M. W. Shah, M. I. Sharif, and F. Marinello, "Comparing inception V3, VGG 16, VGG 19, CNN, and ResNet 50: a case study on early detection of a rice disease," *Agronomy*, vol. 13, no. 6, p. 1633, 2023, doi: https://doi.org/10.3390/agronomy13061633.

[16] G. Beissenova *et al.*, "Using Pretrained VGG19 Model and Image Segmentation for Rice Leaf Disease Classification.," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 8, 2024, doi: 10.14569/ijacsa.2024.0150873.

[17] S. Ponmalar, M. Devavaishnee, and M. K. I. Priya, "Classification of Rice Varieties Using VGG-19 and Alexnet," in *2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, 2023, pp. 1–5. doi: https://doi.org/10.1109/SMARTGENCON60755.2023.10442412.

[18] E. Dhaniswara, Y. Kristian, and E. I. Setiawan, "Detection of Banana and Its Ripeness Using Residual Neural Network," *J. Informatics Telecommun. Eng.*, vol. 5, no. 1, pp. 188–197, 2021, doi: 10.31289/jite.v5i1.4844.

[19] L. Z. A. Mardedi, F. Fahry, M. Madani, and H. Hairani, "Detection of Rice Diseases using Leaf Images with Visual Geometric Group (VGG-19) Architecture and Different Optimizers," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 25, no. 1, pp. 73–82, 2025, doi: https://doi.org/10.30812/matrik.v25i1.5286.

[20] M. Nawaz, T. Nazir, M. A. Khan, V. Rajinikanth, and S. Kadry, "Plant disease classification using VGG-19 based faster-RCNN," in *International Conference on Advances in Computing and Data Sciences*, 2023, pp. 277–289. doi: https://doi.org/10.1007/978-3-031-37940-6_23.

[21] T.-H. Nguyen, T.-N. Nguyen, and B.-V. Ngo, "A VGG-19 model with transfer learning and image segmentation for classification of tomato leaf disease," *AgriEngineering*, vol. 4, no. 4, pp. 871–887, 2022, doi: https://doi.org/10.3390/agriengineering4040056.

[22] N. K. Reddy and M. Rajasekar, "Increasing F1 score with VGG16 during plant disease classification over VGG19," in *AIP Conference Proceedings*, 2024, vol. 3193, no. 1, p. 20191. doi: https://doi.org/10.1063/5.0238123.

[23] P. Y. Yudhis, F. Bimantoro, and R. P. Rassy, "Comparative Analysis of ResNet-50 and VGG16 Architecture Accuracy in Garbage Classification System," *J. Comput. Sci. Informatics Eng.*, vol. 9, no. 1, 2025, doi: https://doi.org/10.29303/jcosine.v8i1.620.

[24] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification," in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, 2021, vol. 1, pp. 96–99. doi: https://doi.org/10.1109/CENTCON52345.2021.9687944.

[25] R. Sujatha, J. M. Chatterjee, N. Z. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocess. Microsyst.*, vol. 80, p. 103615, 2021, doi: https://doi.org/10.1016/j.micpro.2020.103615.